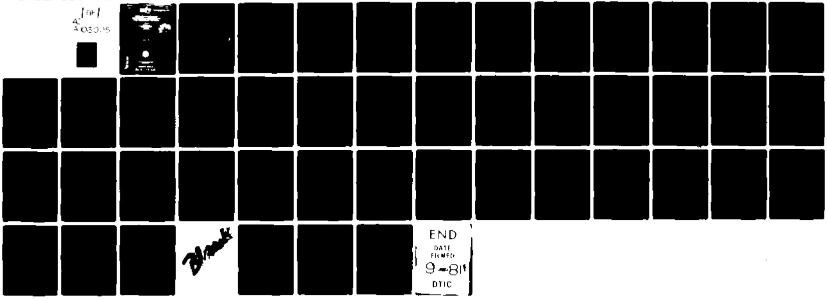
AD-A103 025 NAVAL RESEARCH LAB WASHINGTON DC F/G 20/11
SOLUTION OF TRANSCENDENTAL AND ALGEBRAIC EQUATIONS WITH APPLICA--ETC(U)
AUG 81 C M RUGGIERO
UNCLASSIFIED NRL-MR-4449 NL

1 OF 1
AD
A103025



Blank

END
DATE
FILMED
9-81
DTIC

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NRL Memorandum Report 4449

2. GOVT ACCESSION NO.
AD-A103 025

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
SOLUTION OF TRANSCENDENTAL AND ALGEBRAIC EQUATIONS WITH APPLICATION TO WAVE PROPAGATION IN ELASTIC PLATES

5. TYPE OF REPORT & PERIOD COVERED
Interim report on a continuing work unit.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
C. M. Ruggiero

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Underwater Sound Reference Detachment
Naval Research Laboratory
P.O. Box 8337, Orlando, FL 32856

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61153N-11; RR011-08-42;
0585-0-0

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research Direct Funding Systems Research & Technology Directorate
Naval Research Laboratory, Washington, DC 20375

12. REPORT DATE
August 7, 1981

13. NUMBER OF PAGES
48

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Timoshenko-Mindlin thick-plate theory
Classical plate theory
Dispersion curves
Strain distribution
Iterative rootfinder
Transcendental equations
Nonfluid loaded plates
Poisson's ratio
Lamb modes
Roots of equations
Hewlett Packard computer language
Vibration of plates

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A numerical method is described by which theoretical formulas governing the propagation of symmetric and antisymmetric waves in elastic plate theory may be evaluated. Vital to evaluating the formulas is the availability of a means for solving transcendental equations. A generalized iterative rootfinder that was developed for this purpose is described. This rootfinder provides the investigator having access to computing facilities with a reliable means for solving transcendental equations. Dispersion curves for both the flexural and extensional waves in rubber, steel, and beryllium plates are calculated and plotted.

(Continues)

20. ABSTRACT *(Continued)*

These curves approach a real value κ for the ratio of the Rayleigh to shear wave speeds. Values of this quantity calculated from the exact elasticity theory were compared with those obtained from an approximate formula given by Victorov. In addition, the inflection and the maximum points of the strain distribution throughout a free plate were calculated. It was found that inflection points in the strain distribution do not exist at all frequencies. In addition to a description and flow chart of the iterative rootfinder method, the significant graphs, equations, and computer programs that arose when computing the dispersion curve calculations are included.

CONTENTS

SOLUTION OF TRANSCENDENTAL AND ALGEBRAIC EQUATIONS WITH APPLICATION TO WAVE PROPAGATION IN ELASTIC PLATES

INTRODUCTION

Transcendental equations with the general form

$$\frac{A(x)}{B(x)} = \frac{C(x)}{D(x)} \tag{1}$$

arise in elastic plate theory as the characteristic equation governing the propagation of symmetric and antisymmetric waves. This report documents the development of a generalized iterative rootfinder for the solution of this type of equation and its application to elastic plate theory. This report, therefore, represents a dichotomy of topics. The initial portion of the text gives a thorough development of the generalized rootfinder used in solving transcendental equations in the form of Eq. (1). The latter part applies the generalized method to solve a number of equations arising in elastic plate theory.

The organization of this paper is as follows: First, the approximation of a root of the equation $f(x) = 0$ by a known iterative method is reviewed. A simple transcendental equation is then chosen to exemplify this basic method. Next, the generalized iterative rootfinder is introduced. From examining this generalized method, the reader will see how the basic iterative scheme reviewed previously can be successively applied to determine the roots of the general equation with the form of Eq. (1). Using this general iterative rootfinder, one can calculate and plot dispersion curves for both the

Manuscript submitted May 22, 1981.

flexural and extensional waves in rubber, steel, and beryllium plates. These curves are found to approach a real value κ (ratio of the Rayleigh to shear wave speed) for the two fundamental modes of plate vibration. The values of κ calculated from exact elasticity theory are compared with those obtained from an approximate formula by Victorov. In addition, the inflection and maximum points of the strain distribution throughout a free plate were calculated. Finally, the strain distribution throughout a free plate was determined and results presented.

ITERATIVE ROOTFINDER

Basic Method

Suppose F(x) is a real continuous function such that

$$F(x) = 0 \tag{2}$$

has real roots. In general, these real roots can be computed to any degree of accuracy by an iterative procedure. The iterative method used is based on rewriting Eq. (2) in the form

$$f(x) = g(x). \tag{3}$$

Let z be an actual root of Eq. (3). Take x_0 as the initial approximation of z. (This initial trial root x_0 will subsequently be referred to as the "seed" for the iterative calculation.) If x is replaced by x_0 in the right-hand side of Eq. (3), then

$$f(x) = g(x_0), \tag{4}$$

which can be solved by calculating x_1 in the form

$$x_1 = f^{-1}[g(x_0)]. \tag{5}$$

The value x_1 obtained by Eq. (5) becomes the argument of a new approximation

$$x_2 = f^{-1}[g(x_1)], \tag{6}$$

where the value x that satisfies Eq. (6) is x_2. In general, the n^{th} approximation x_n is determined by solving

$$x_n = f^{-1}[g(x_{n-1})]. \tag{7a}$$

If the sequence x_0, x_1, x_2 ... tends to a limit z and if f(x) and g(x) are continuous at the point z, then z is the root of the given equation.

One might suppose that Eq. (5) could have been written as

$$x_1 = g^{-1}[f(x_0)], \tag{7b}$$

instead of Eq. (7a). However, the side of Eq. (3) that needs to remain fixed by the insertion of x_0 is not arbitrary. If the two functions f(x) and g(x) have first derivatives such that

$$|f'(x)| > |g'(x)|, \tag{8}$$

then the sequence x_0, x_1, x_2 ... is convergent whenever written in the form of Eq. (7a), but not when written in the form of Eq. (7b). That is, the function with the smaller slope in the neighborhood of z of Eq. (3) must be fixed in order to obtain successively better approximations of the true root z.

Sokolnikoff and Redheffer [1] describe the geometric interpretation of this iterative procedure. Equation (3) describes two curves, namely $y = f(x)$ and $y = g(x)$, which intersect at the root z. If the slopes of these two curves have the same sign and satisfy Eq. (8), the convergence is in a step-like pattern as shown in Fig. 1. One may note that in this case

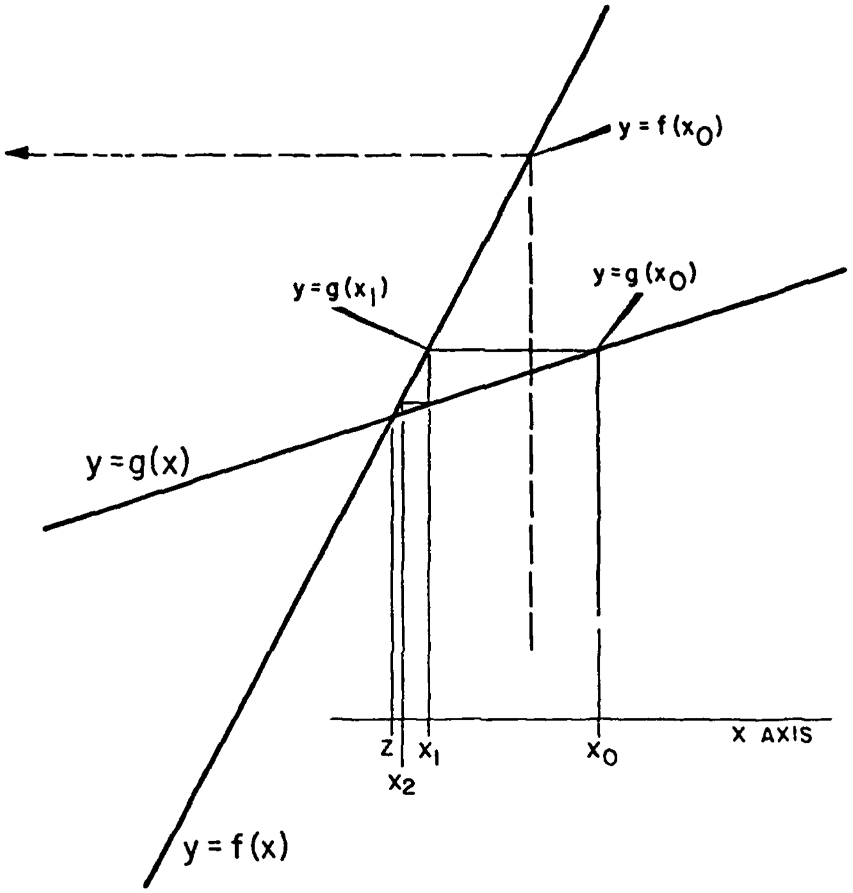


Fig. 1. Intersection of two equations having slopes with the same sign. Solid line is result of fixing equation with smaller slope. Dashed line indicates a convergence as a result of choosing the larger slope.

successive approximations approach the true root z from one side. On the other hand, when the slopes of the curves have opposite signs, as shown in

Fig. 2, convergence has a spiral-type configuration. The successive approximations in Fig. 2 oscillate on both sides of z and yet approach the true root z when the curve g(x), with the smaller slope is fixed. The dotted line shows how the iterative process can generate approximations that diverge from z. This occurs when f(x), the curve with the larger slope, is fixed rather than g(x). Regardless of whether convergence takes place in a stepwise or a spiral fashion, the curve with the smaller slope must be the one fixed in Eq. (3), otherwise consecutive approximations will not approach z but instead diverge from it.

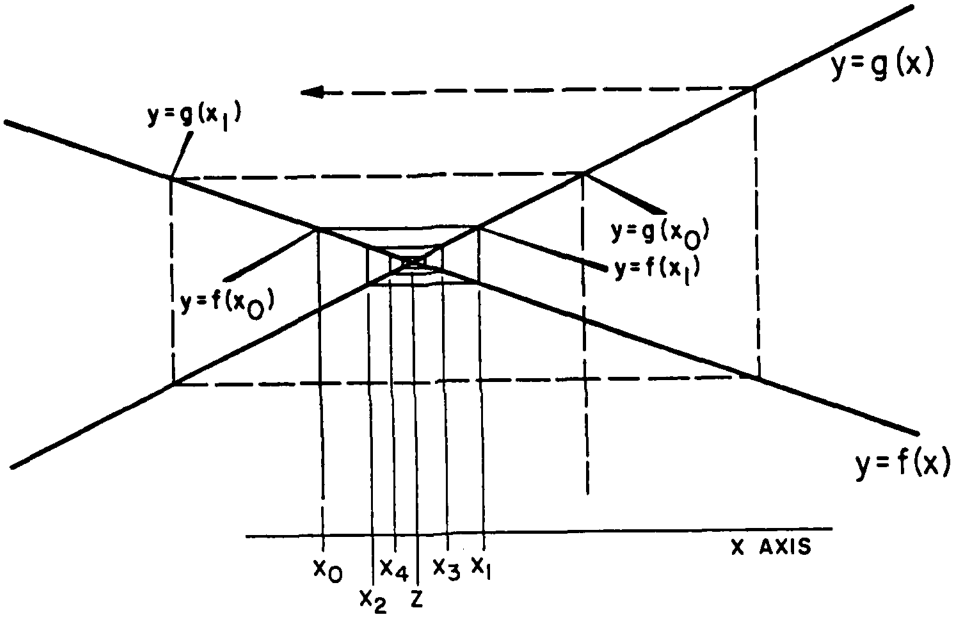


Fig. 2. Intersection of two equations having slopes of opposite sign.

Since with any numerical method, exact roots cannot be found, a degree of accuracy must be decided on. Therefore, whenever the difference $|x_n - x_{n-1}|$

is within a predetermined tolerance, x_n is accepted as a sufficiently good approximation of z, the true root of Eq. (2).

Example

It seems appropriate to examine at this time an example of this iterative scheme [2]. Suppose one wishes to determine the roots of

$$\tan(x) - \frac{2}{x} = 0. \tag{9}$$

The real roots of this equation are the absissas of the point of intersection of the curves

$$y = \tan(x) \tag{10a}$$

and

$$y = \frac{2}{x} . \tag{10b}$$

A trial root or seed x_0 should be in the vicinity of the true root for a more rapid convergence. However, with the use of the computer, the extra time required when x_0 is not chosen near the actual root z is usually negligible.

The first trial root x_0 may be chosen to be equal to one. The slope of y = tan x is greater than that of $y = \frac{2}{x}$ at x = 1.0. Therefore, Eq. (9) is written in the form

$$x_1 = \tan^{-1}(2/x_0) \tag{11a}$$

or in general, for the n^{th} iteration

$$x_n = \tan^{-1}\left(\frac{2}{x_{n-1}}\right). \tag{11b}$$

The first seven terms of the sequence generated by Eq. (11b) with $x_0 = 1.0$, are $x_1 = 1.1071$, $x_2 = 1.0652$, $x_3 = 1.0814$, $x_4 = 1.0751$, $x_5 = 1.0775$,

$x_6 = 1.0766$, and $x_7 = 1.0769$. The computations can be terminated at this stage if the predetermined tolerance t is set, say, at 3×10^{-4} because $|x_6 - x_5| \leq t$. It is important to point out that the tolerance is not necessarily the error between the n^{th} computed root x_n and the true root z, as indicated in many books (see Appendix A for a detailed analysis of the error and tolerance size).

It should also be noted that if one function in Eq. (3) has a larger slope at one point, this does not ensure that it will have the larger slope at every point. For instance, in the example considered here, the derivative of 2/x is $-2/x^2$ and the derivative of tan(x) is $[\cos(x)]^{-2}$. When $x = 1$, the slope of 2/x is -2 while the slope of tan(x) is 3.425. Therefore, $\frac{d}{dx}\tan(x) > \frac{d}{dx}\left(\frac{2}{x}\right)$ at $x = 1$. However, if instead 0.1 were chosen as the value of x, the slope of 2/x is -200 while the slope of tan(x) is 1.005. Thus, one would find that $\frac{d}{dx}\left(\frac{2}{x}\right) > \frac{d}{dx}\tan(x)$ at $x = 0.1$. Hence by Eq. (8), Eq. (9) should be written as

$$x_1 = 2\tan(x_0) \tag{11c}$$

rather than in the form of Eq. (11c). However, Eq. (11c) will produce successive roots that diverge from the true root. Therefore, it is important that the initial trial root x_0 be close enough to the true root so that the slopes of the function f and g in Eq. (3) in the neighborhood of the trial root will have the same relation to one another as they have in the neighborhood of the true root.

Generalized Method

The basic method discussed above is effective when an equation can be written in the form $f(x) = g(x)$. However, for an equation written in the form

$$\frac{A(x)}{B(x)} = \frac{C(x)}{D(x)}, \tag{12a}$$

the iterative rootfinder described above must be made more general. In the generalized method, the basic iterative method described above is successively applied to combinations of the functions that appear in Eq. (12) until the root is determined. The rationale for solving Eq. (12a) is as follows:

Let $\frac{A(x)}{B(x)} = F(x)$ and $\frac{C(x)}{D(x)} = G(x)$, and suppose the slopes of F(x) and G(x) can be determined. If x_0 is the first trial root and F(x) has the greater slope at x_0, then Eq. (12a) can be written in the form

$$\frac{A(x)}{B(x)} = G(x_0). \tag{12b}$$

Equation (12b) can also be written as

$$A(x) = G(x_0)B(x), \tag{13a}$$

where $G(x_0)$ serves as a constant since G is evaluated at x_0. Note that if G(x) had a slope larger than F(x) at x_0, Eq. (12a) would have been written as

$$C(x) = F(x_0)D(x). \tag{13b}$$

The two pairs of functions, $y = A(x)$ and $y = G(x_0)B(x)$ of Eq. (12a), or $y = C(x)$ and $y = F(x_0)D(x)$ of Eq. (13b), also demand a slope comparison to assure that successive approximations will approach the true root z. Consequently for an equation in the form of (12a), two slope comparisons need to be made for each approximation. To deal with an equation in this form, a "Little Loop-Big Loop (LL-BL) Procedure" was designed. The "Little Loop-Big Loop Procedure" breaks down Eq. (12a) into the form of Eq. (3). Basically this is done by making two slope comparisons and implementing the iterative procedure.

It should be kept in mind that this LL-BL procedure works not only for equations of the form of Eq. (12a), but can also be implemented for equations involving three functions, $\frac{A(x)}{B(x)} = C(x)$, two functions $A(x) = B(x)$, or even one function such as $A(x) = C$ where C is a constant.

First the slopes of F(x) and G(x) are compared in the BL. Then by the criterion given in Eq. (8), Eq. (12a) will be written in the form of either Eq. (13a) or Eq. (13b) and will be passed to the LL. In the LL, the slopes of the functions of Eq. (13) will be calculated. Again by the criterion given in Eq. (8), one of the following equations will result. If Eq. (13a) was used, then one will have

$$x = A^{-1}[B(x)\ G(x_0)] \tag{14a}$$

when

$$|[G(x_0)B(x)]'| < |A(x)'|$$

or

$$x = B^{-1}[A(x)/G(x_0)] \tag{14b}$$

when

$$|[G(x_0)B(x)]'| > |A(x)'|,$$

and, if Eq. (12b) were used,

$$x = C^{-1}[F(x_0)D(x)] \tag{14c}$$

when

$$|[F(x_0)D(x)]'| < |C(x)'|$$

or

$$x = D^{-1}[C(x)/F(x_0)] \tag{14d}$$

when

$$|[F(x_0)D(x)]'| > |C(x)'| .$$

As seen above, there are four possible ways to rewrite Eq. (12a); however, there is only one equation that results in convergent approximations as discussed earlier in the basic iterative procedure.

After the proper version of Eq. (14) is chosen, the difference between successive approximations must be made less than a preset tolerance. This tolerance is called the LL-tolerance. The root r_1 that satisfies the LL tolerance is then recirculated into Eq. (12a) to again make the BL slope comparison. After the proper forms of Eqs. (13) and (14) have been chosen for trial root r_1, the iterative procedure is carried out and a new root r_2 results, where r_1 replaces x_0. In general, after n such "recirculations", when $|r_{n-1} - r_n|$ is less than a preset tolerance, r_n is taken to be the root of Eq. (12a) and this tolerance is called the BL tolerance. Note there is both a LL tolerance and a BL tolerance, and these need not be the same. In practice, a LL tolerance smaller than the BL tolerance is chosen. However, this choice, which presently worked in the application of the generalized iterative method, lacks a rigorous rationale. An investigation of how each of the tolerances should be set would give further insight in this area.

There is not a one-to-one correspondence between the number of iterations in the LL and those of the BL. In fact, there may be anywhere from one to several hundred iterations in the LL for each iteration in the BL. The flowchart in Appendix A outlines the logic of this LL-BL scheme for solving transcendental equations. After all desired roots of Eq. (12a) are found, they are stored in an array.

DISPERSION CURVES FOR ANTISYMMETRIC AND SYMMETRIC WAVES

The algorithm for determining roots outlined above will now be applied to the calculation of the dispersion curves [3] of the exact theory, which can be suitably rewritten for the purpose of these calculations in the following dimensionless forms.

For antisymmetrical modes, the equation for the dispersion curves is

$$\frac{(1 - \alpha\gamma^2)(1 - \gamma^2)^{\frac{1}{2}}}{\tanh[\frac{1}{2}x(1 - \alpha\gamma^2)^{\frac{1}{2}}]} = \frac{(1 - \frac{1}{2}\gamma^2)^2}{\tanh[\frac{1}{2}(1 - \gamma^2)^{\frac{1}{2}}]} , \tag{15}$$

which holds when $0 < \gamma^2 < 1$, and for symmetric modes the equation is

$$\frac{(1 - \alpha\gamma^2)^{\frac{1}{2}}(\gamma^2 - 1)^{\frac{1}{2}}}{\tan[\frac{1}{2}x(\gamma^2 - 1)^{\frac{1}{2}}]} = \frac{(1 - \frac{1}{2}\gamma^2)^2}{\tanh[\frac{1}{2}x(1 - \alpha\gamma^2)^{\frac{1}{2}}]} , \tag{16}$$

which holds when $1 < \gamma^2 < \frac{1}{\alpha}$, and

$$\frac{(1 - \frac{1}{2}\gamma^2)^2}{\tanh[\frac{1}{2}x(1 - \alpha\gamma^2)^{\frac{1}{2}}} = \frac{(1 - \alpha\gamma^2)^{\frac{1}{2}}(1 - \gamma^2)^{\frac{1}{2}}}{\tanh[\frac{1}{2}x(1 - \gamma^2)^{\frac{1}{2}}]} , \tag{17}$$

which holds when $\gamma^2 < 1$. In Eqs. (15) through (17), the normalized wave speed γ is a function of $x = \pi fd/c$, where c, f, and d are, respectively, the wave speed in the plate, the frequency, and one-half the thickness of the plate.

In Eqs. (15) through (17), α is expressed in terms of Poisson's ratio ν of the plate material by the equation

$$\alpha = \frac{1 - 2\nu}{2(1 - \nu)} \quad . \tag{18}$$

It should be noted that in the dispersion equations for the lowest order symmetric Lamb mode [(Eqs. (16) and (17)], the value γ^2 never exceeds $1/\alpha$ since $0 \leq \nu \leq 0.5$. However, in higher modes, γ^2 does exceed 2.0, thus making $1/\alpha$ an important consideration.

A computer program was designed so that it could easily be implemented for various equations with a subprogram containing the functions and their inverses independent of the LL-BL iterative scheme. In the subprogram, parameters are represented by p-numbers (parameter numbers). These p-numbers [4] are unique to the computer (HP 9825) language that was used. This enables one to call the subprogram repeatedly using different values for the parameters each time. The dispersion curves for antisymmetrical waves are shown in Fig. 3 for values of Poisson's ratio equal to 0.5 (rubber), 0.303 (steel), and 0.05 (beryllium).

In order to calculate these dispersion curves, Eq. (15) must be broken down into four functions shown in Eq. (12a). The function assignment chosen for Eq. (15) is $A = (1 - \alpha\gamma^2)(1 - \gamma^2)^{\frac{1}{2}}$, $B = \tanh[\frac{1}{2}x(1 - \alpha\gamma^2)^{\frac{1}{2}}]$, $C = (1 - \frac{1}{2}\gamma^2)^2$, and $D = \tanh[\frac{1}{2}(1 - \gamma^2)^{\frac{1}{2}}]$. The functions do not have to be labeled this way; but, whatever choice is made, A/B must equal C/D. Appendix (D) illustrates the subprogram for calculating the dispersion curves associated with the first order antisymmetric Lamb mode. The expressions AA, BB, CC, and DD are the inverses of the corresponding functions A, B, C, and D. Since both tanh and arctanh are needed for this set of functions and since the computer

(Hewlett-Packard 9825) does not have written algorithms for these functions, appropriate subroutines were written and included in this subprogram.

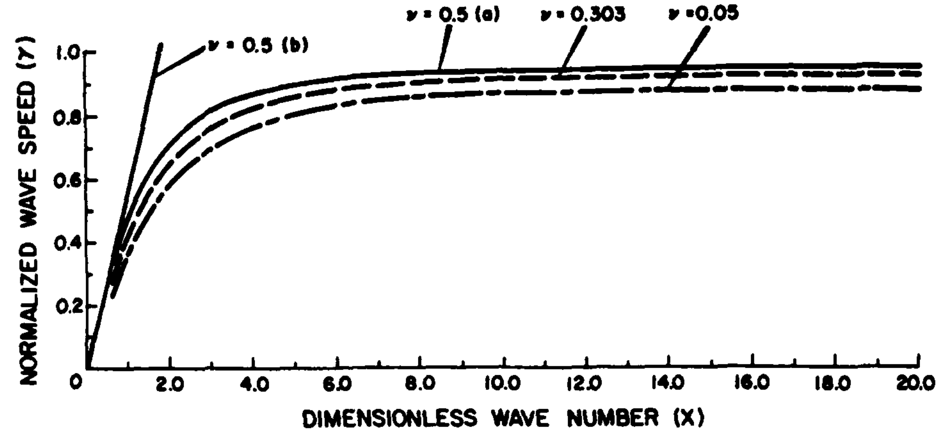


Fig. 3 - Antisymmetric flexural waves of rubber ($\nu = 0.5$), steel ($\nu = 0.303$), and beryllium ($\nu = 0.05$).

Curve a is the flexural wave of thick-plate theory.

Curve b is the flexural wave of classical plate theory for the same material.

The iterative program (see Appendix C) using the set of functions A, B, C, and D given above, performs adequately for values of $x \geq 0.4$. For values of $x < 0.4$, however, excessive round-off errors were introduced. To circumvent these errors, the classical-plate-theory equation was used for values of $x < 0.4$. The equation for this dispersion curve is

$$\gamma = (\tfrac{1}{2}x)\,\frac{2}{3(1-\nu)}\ . \qquad (19)$$

This expression gives a dispersion curve that is a straight line, as shown by curve (b) of Fig. 3. Curve (a) is the dispersion curve given by the exact theory. Thus, as Fig. 3 illustrates, Eq. (19) is a good approximation of the thick-plate-theory for $x \leq 0.4$. When calculating a dispersion curve for antisymmetric waves, the old root was used as the new seed.

Each dispersion curve for symmetrical waves (see Fig. 4) requires two sets of functions. One set of functions is entered when $1 < \gamma^2 < 1/\alpha$ and another set of functions when $\gamma^2 < 1$. Equations (16) and (17) are the characteristic equations for these dispersion curves. To calculate the curves shown in Fig. 4, it was necessary to overcome several difficulties.

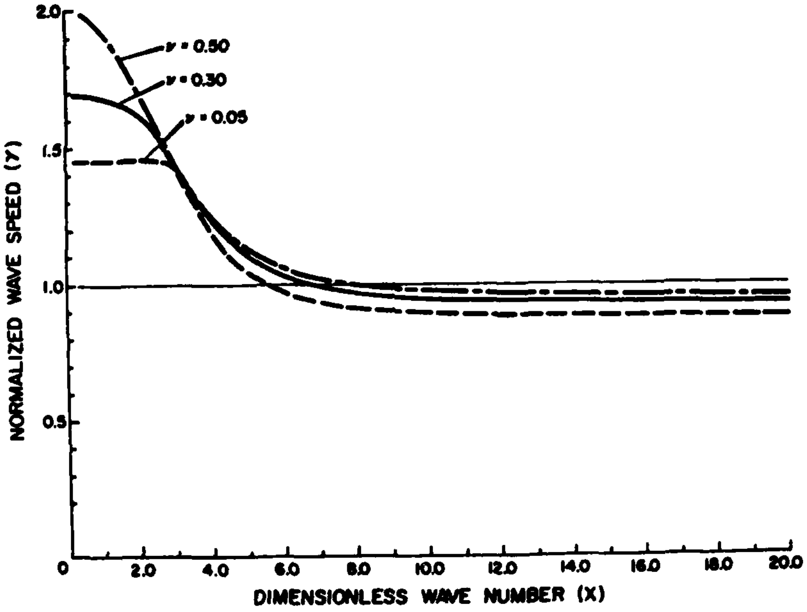


Fig. 4. Symmetrical compression waves of rubber ($\nu = 0.5$), steel ($\nu = 0.303$), and beryllium ($\nu = 0.05$).
The line $\nu = 1$ is a critical value.

This was done by using a seed that is a better approximation of the true root z; rather than using a root from the previous x as a seed for current x as was done in antisymmetric case [see the example illustrated using Eq. (9)]. By linearly extrapolating from the old roots, one could calculate a a root γ_n for any x using the iterative rootfinder. To determine a seed γ_0 for current calculations of γ_n, one must compute $\Delta\gamma$. The expression $\Delta\gamma$ is given as

$$\Delta\gamma = \gamma_{n-2} - \gamma_{n-1} ,$$

where γ_{n-1} is the root associated with the previous valued x and γ_{n-2} is that root calculated before γ_{n-1}. To compute γ_0 for the symmetric mode, the following equation is used,

$$\gamma_0 = \gamma_{n-1} - \Delta\gamma \quad .$$

The algorithm for performing the extrapolation can be found in Appendix C in the subroutine 'DO'. It has been shown that as x increases, 'DO' appropriately adjusts f(x) so that the desired root is calculated. The several difficulties that were overcome by the extrapolation will be explained in detail.

The first difficulty arose when it was necessary to change from one functional form to another at $\gamma = 1$ (see Fig. 4). The dispersion curves for the symmetrical waves are calculated by using Eq. (16) when $\gamma < 1$ and using Eq. (17) when $1 < \gamma < \alpha^{-\frac{1}{2}}$. However, if a value of γ that is greater than 1 is used for the seed of the iterative scheme at an x value where the true γ should be less than 1, the computer program will always work with the particular functions that appear as A, B, C, and D in Eq. (16). These functions will always produce roots γ greater than 1. The iterative program, therefore, will never get to use the set of functions that appear in Eq. (17), which are needed to obtain a value of γ less than 1. So the iteration cannot be started with a seed $\gamma > 1$ if it is to produce a root $\gamma < 1$. The extrapolation remedied this problem. It reduced the previously obtained value of γ enough so that the seed needed to compute values of $\gamma < 1$, such that γ would itself be less than 1 at the appropriate x, and hence allowed the new set of functions to be entered.

Another difficulty was revealed when the absolute value of the slopes of $\frac{A(x)}{B(x)}$ and $\frac{C(x)}{D(x)}$ were calculated using the root obtained for the previous value of x as the seed for the calculation with an incremented x. If the derivatives of A/B and C/D are nearly equal in magnitude, the value chosen for the seed could make a significant difference as to which side of Eq. (12a) is fixed. If the wrong set of functions [(i.e., the wrong side of Eq. (12a)] is fixed in the BL, then the LL calculations cannot converge to the true root, and the LL iterations will continue indefinitely. For example, if Eq. (13a) were incorrectly chosen by the program instead of Eq. (13b) in the BL, the LL would be a root of Eq. (14a) and (14b) without converging to a definite value that would return the calculation to the BL. However, if the extrapolation procedure is used to determine the seed, the BL slope comparison is more likely to identify the correct side of Eq. (12a) to fix, since the seed obtained by extrapolation is closer to the true root than is the value of γ obtained on the previous calculation.

And the third problem was exhibited at values of x where the dispersion curve expressed by Eq. (16) and the curve given by

$$\gamma^2 = + (\pi/x)^2 \tag{20}$$

pass close to one another. Equation (20) gives the locus of values for which the argument of the tangent in the equation

$$B(x) = \tan[\tfrac{1}{2}x(\gamma^2 - 1)^{\frac{1}{2}}] \tag{21}$$

is equal to $\pi/2$; that is, it gives values of γ for which $\frac{1}{2}x(\gamma^2 - 1)^{\frac{1}{2}} = \pi/2$. The tangent is not continuous at the point $x = \pi/2$; one has

$$\lim_{t\to\left(\frac{1\pi}{2}\right)^+} \tan(t) = -\infty , \qquad \text{and} \qquad \lim_{t\to\left(\frac{1\pi}{2}\right)^-} \tan(t) = +\infty$$

where

$$t = \tfrac{1}{2}x(\gamma^2 - 1)^{\frac{1}{2}} \quad .$$

It is important not to allow the iterative calculation procedure to encounter this point of discontinuity. For example, suppose x has been progressively incremented and many roots have been calculated from Eq. (16) with $\gamma^2 < 1 +(\frac{\pi}{x})^2$.

If the next increment of x is such that

$$\gamma^2 > 1 + (\frac{\pi}{x})^2 ,$$

for the seed used, then a root γ_0 will be produced that involves a different branch of the inverse tangent function than that which was used in the calculation of previous points on the dispersion curve. If this occurs, the computing algorithm will produce an undesired root. The extrapolation procedure, however, keeps $\gamma^2 < 1 + (\frac{\pi}{x})^2$ by subtracting an appropriate amount from the γ associated with the previous value of x and one always uses the correct branch of the inverse tangent function.

It has been shown all the difficulties encountered in determining the dispersion curves for symmetric waves were overcome by linearly extrapolating. The iterative program works for a wide range of values of Poisson's ratio; however, for small values of this quantity ($\nu \leq 0.05$) additional program modifications were necessary. These modifications include choosing a seed for each x that is less than $1 + (\pi/x)^2$. This was accomplished by taking the extrapolated seed e and checking if $e < 1 + (\pi/x)^2$. If so, e is returned as the seed for the new iteration. However, if $e > 1 + (\pi/x)^2$, the quantity $1 + (\pi/x)^2$ will become the new seed instead of e. This subroutine is called "TN" and may be found in Appendix E. Also, if the similar-slope problem previously encountered is not resolved by the extrapolation described above, further control is needed to determine which set of functions in the BL should be fixed. This was accomplished by fixing one set of functions in the BL and iterate in the LL until a root was outputted. However, if a divergence exists in successive approximations, the wrong set of functions was fixed in the BL. Therefore, the other set of functions would be fixed. A divergence is assumed when the number of LL iterations exceeds 500. The number 500 is arbitrary, and possibly a larger number would be needed for some value of Poisson's ratio. These changes are also found in the program in Appendix E.

This revised program worked for beryllium ($\nu = 0.05$) and is believed to work for $\nu < 0.05$. This alternative iterative program can be found in Appendix E as a supplement to the original program in Appendix C. However, the modification of the program produces the roots much more slowly. For this reason, it is recommended that one use the program in Appendix C unless they are using a value of Poisson's ratio less than 0.10.

ADDITONAL CALCULATIONS

Kappas (κ)

Figure 5 illustrates dispersion curves for both the flexural and compressional waves of a rubber material ($\nu = 0.5$). Both curves approach a real value $\kappa_R = c_R/c_s$ where c_R and c_s are, respectively, the Rayleigh wave speed and the shear wave speed.

It was of interest to graphically illustrate the behavior of κ^2 as a function of Poisson's ratio when κ is computed by two different methods.

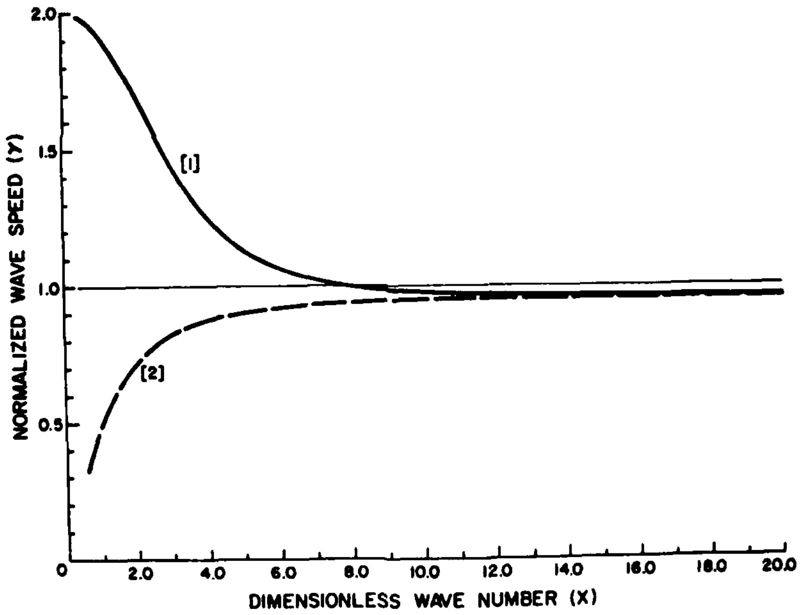


Fig. 5. Symmetrical [1] and antisymmetrical [2] waves of rubber. Kappa is the point where the two curves approach the normalized Rayleigh wave velocity

Note that as α changes from 0 to 0.5, Poisson's ratio ν changes from 0.5 to 0. Exact elasticity theory, κ gives as a real root of the equation

$$\kappa^6 - 8\kappa^4 + 8(3 - 2\alpha)\kappa^2 - 16(1 - \alpha) = 0$$

where α is defined in Eq. (18). The real root of this equation is plotted as the solid curve in Fig. 6. In the manuafacturer-supplied software package for the HP 9825 computer, there was a "Polynomial Rootfinder" program that

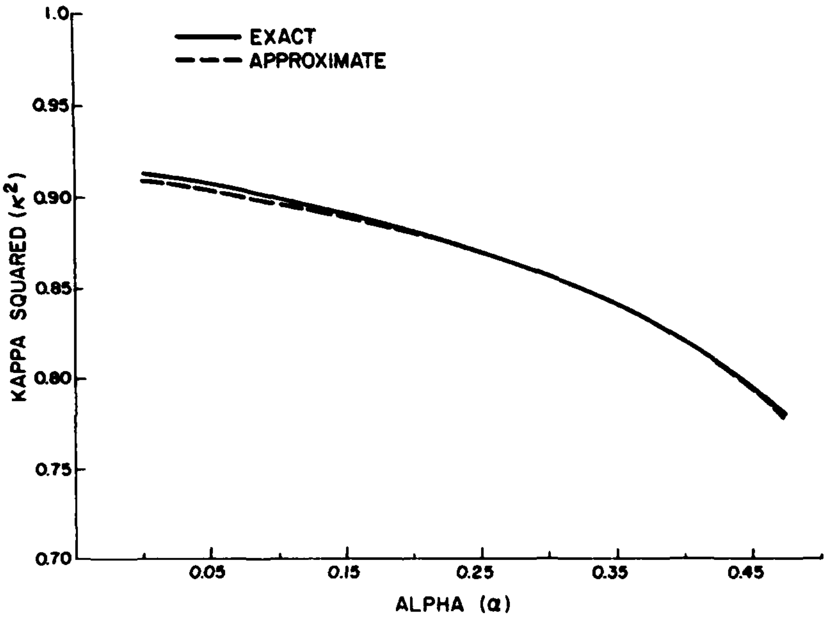


Fig. 6. Viktorov's approximation of κ^2 compared with the exact theory value for κ^2 as a function of α.

could determine the roots of such polynomial equations. However, in that program, the coefficients of the polynomial must be manually entered and the output will consist of all the roots (both real and complex). To calculate κ as a function of the variable α, it was necessary to have the computer determine the coefficients and choose only the real roots.

The approximation of κ proposed by Viktorov [5] is given by

$$\kappa^2 = \left(\frac{0.87 + 1.12}{\nu + 1}\right)^2 \quad .$$

By calculating κ as a function of α using the results of Eq. (18), one obtains the dashed curve in Fig. 6. Note the agreement between the two curves in Fig. 6.

Inflection Points

The quantity κ has an important role in approximate plate theory. Dr. P. Dubbelday is investigating improved expressions for κ. In this investigation, it was necessary to examine features of the strain distribution for antisymmetric waves. The formula for the strain distribution produced by antisymmetric waves is

$$\varepsilon_{zx} = \cosh[qd(z/d)]\cosh(sd) - \cosh[sd(z/d)]\cosh(qd) \tag{22}$$

where z is the distance from center of the plate and d is half the plate thickness. The expressions qd and sd are given by the equations $qd = x(1 - \alpha\gamma^2)^{\frac{1}{2}}$ and $sd = x(1 - \gamma^2)^{\frac{1}{2}}$ whereas previously $x = (\pi d)/\lambda$ with λ the wavelength of the antisymmetric wave. In particular, it was believed that the inflection point of Eq. (22) should be an identifiable point of the curve that merges with the center of the plate at the low-frequency limit. When the derivative ε'_{zx} is equal to zero, we obtain a maximum or minimum; and when the second derivative ε''_{zx} is equal to zero, we get an inflection point. The inflection points (kz_i) are given in the form $(qd)^2\cosh[kz_i(1 - \alpha\gamma^2)^{\frac{1}{2}}]\cosh(sd) = (sd)^2\cosh[kz_i(1 - \alpha\gamma^2)^{\frac{1}{2}}]\cosh(qd)$. Figure 7 illustrates the location of the

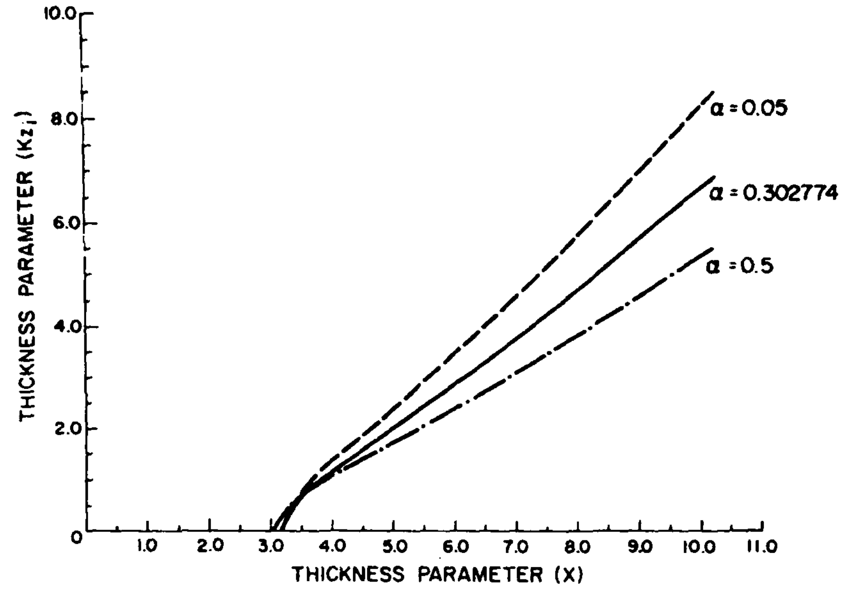


Fig. 7. Point of inflection for rubber (0.5), steel (0.302774), and beryllium (0.05).

inflection point in the plate with respect to the plate's center as a function of x for three materials, steel (ν = 0.302774), rubber (ν = 0.5), and beryllium (ν = 0.05). It was originally believed that these inflection points were a characteristic feature that could be used in the theory being developed for calculating κ. However, the graph reveals the fact that points of inflection do not exist for values of x less than approximately 3.2. Thus, the hypothesized method of finding κ could not be used.

Maximum Points

Finding the point of maximum strain kz_m parallels the findings of inflection points. The maximum points kz_m are given as the root of

$$(qd)\sinh[kz_m(1 - \alpha\gamma^2)^{\frac{1}{2}}] = (pd)\sinh[kz_m(1 - \gamma^2)^{\frac{1}{2}}].$$

Figure 8 illustrates the location of points of maximum for rubber and beryllium.

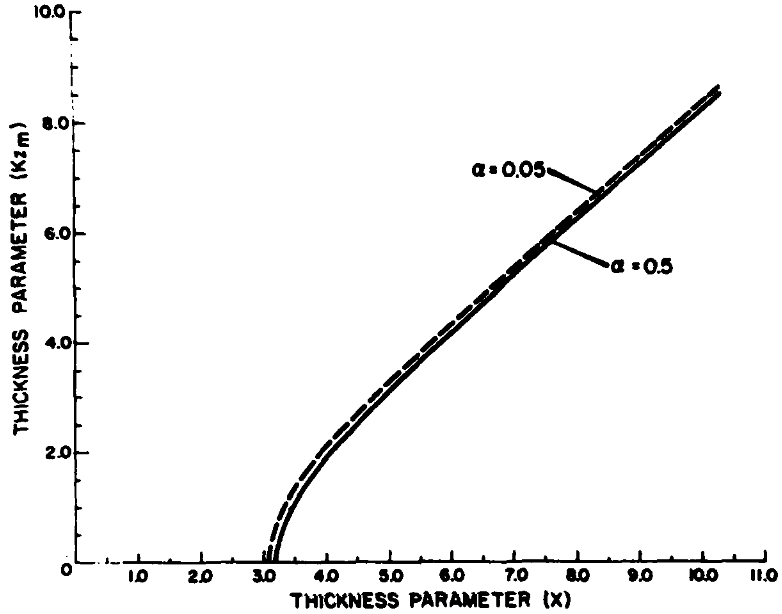


Fig. 8. Point of maximum for rubber (0.5) and beryllium (0.05). Note that kz_m is zero for values of x less than approx. 3.2.

The point of maximum and the point of inflection intercept the x axis at approximately the same point for each material; however, unlike the inflection curve, the values of the maximum curve exists for all x. They are zero for x less than 3.2.

Strain Distribution

After performing the calculations to find points of the inflection and maximum in the strain component that is given by Eq. (22); the normalized strain distribution $\epsilon_{zx}/\epsilon_{zx(m)}$ for distinct values of x was calculated.

The formula for this normalized strain distribution is $\varepsilon_{zx}/\varepsilon_{zx(m)}$, where

$$\varepsilon_{zx(m)} = \cosh(qz_m)\cosh(sd) - \cosh(sz_m)\cosh(qd)$$

and ε_{zx} is given in Eq. (21). The value qz_m and sz_m are given by the following equations, $qz_m = kz_m(1 - \alpha\gamma^2)^{\frac{1}{2}}$ and $sz_m = kz_m(1 - \gamma^2)^{\frac{1}{2}}$. Figure 9 illustrates the strain distribution for various values of x. Note that this quantity has been plotted so that z/d is the ordinate and the strain distribution is the abscissa. It was done in this manner to best reflect the physical situation by a propagating antisymmetric Lamb wave. Note when x is less than about 3.2, the transverse shear strain is 1 in a region near the center of the plate. As x increases above 3.2, the shear strain in this region decreases for values and the position of maximum strain moves towards the surface of the plate.

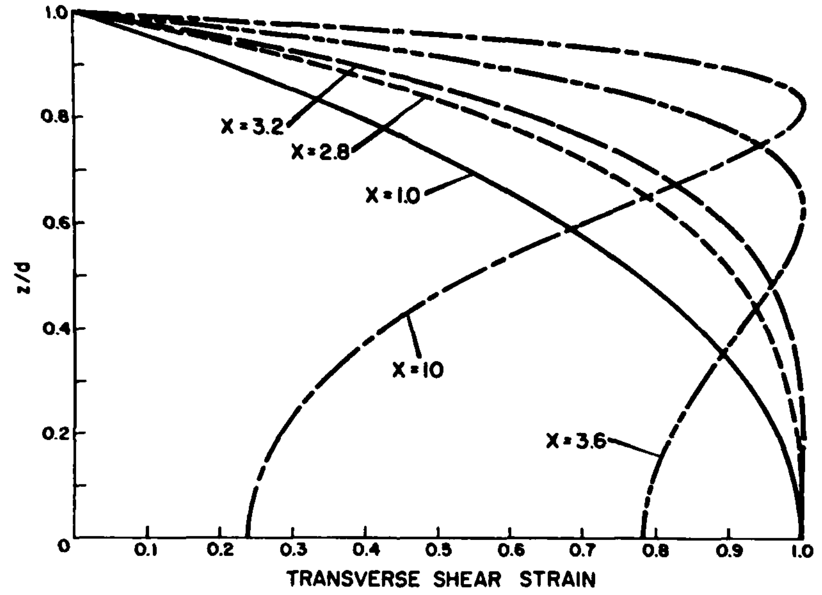


Fig. 9. Strain distribution of the exact theory with $\alpha = 0.302774$.

The strain distribution is plotted in Fig. 10 for three materials when x has a value of 10. Figure 10 shows that the normalized strain is quite different at the center of the plate and yet much the same near the surface. Similar results were found when kd assumed values other than 10.

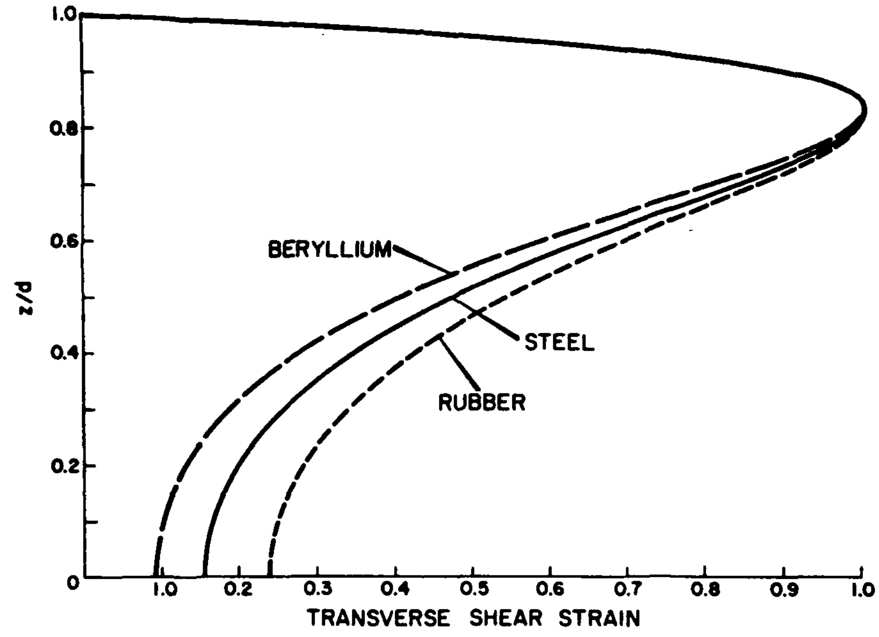


Fig. 10. Comparison of the strain distribution of three materials with x = 10.

SUMMARY

The roots of transcendental equations that arise in elastic plate theory were calculated using a generalization of a well known iterative method. The basic iterative method is successively applied to combinations of functions appearing in the transcendental equations until the root with the desired tolerance is determined. The generalized rootfinder described is a straightforward technique for solving transcendental or algebraic equations. This generalized scheme was applied to the exact elasticity theory dispersion equation, the polynomial equations for calculating κ, and the equations that describe points of strain inflection and strain maximum. It was found that inflection points in the strain distribution do not exist at all frequencies. Since the time this paper was drafted, several other research projects at USRD have implemented this generalized method in their numerical analysis.

For this reason, it appears that other investigators, particularly those working in elasticity theory and in the area of numerical analysis should be able to make use of the iterative rootfinding method with minimal difficulty.

ACKNOWLEDGMENTS

The author gratefully acknowledges Anthony J. Rudgers for his continuing expert guidance, and for critically checking the entire manuscript and offering improvements on many points. She also wishes to thank Pieter W. Dubbelday for his valuable comments and assistance with the BL-LL iterative rootfinder. Finally, thanks are due to Aileen Beard for her patient and expert typing help; to Carol Shuler for preparing the graphics; and to Robert P. Carnegie for his help in photographing the figures for this report.

REFERENCES

[I] I. S. Sokolnikoff and R. M. Redheffer, Mathematics of Physics and Modern Engineering (McGraw-Hill, New York, 1958) pp. 679-684.

[2] Survey of Applicable Mathematics, edited by Karel Rektorys (M.I.T. Press, Massachusetts, 1969) pp. 1179, 1180.

[3] R. D. Mindlin, "Influence of Rotatory Inertia and Shear on Flexural Motions of Isotropic, Elastic Plates," J. Appl. Mech. 18, 31-38 (1951).

[4] Hewlett-Packard 9825A Calculator Advanced Programming (Hewlett-Packard Company, Fort Collins, 1976) p. 19.

[5] I. A. Viktorov, Rayleigh and Lamb Waves (Plenum, New York, 1967) pp. 72-75.

REFERENCES NOT CITED

Pieter S. Dubbelday, "Effective Shear Modulus for Flexural and Extensional Waves in an Unloaded Thick Plate," NRL Report 8372 (submitted for publication).

Anthony J. Rudgers, "Acoustical reflection and radiation by thick fluid-loaded composite plates", Acous. Soc. Am. 66, 571-578 (1979).

APPENDIX A

TOLERANCE AND ERROR SIZE

It is noted that the difference between two successive approximate roots of an equation is not necessarily a measure of the error between the approximate and the true root of the equation. The tolerance will be defined as the difference between successive approximations and the error size defined as the difference between the approximate and true root. The iterative method described in the text is used to solve an equation in the form

$$f(x) = g(x). \tag{A1}$$

It will be shown that the tolerance and error size can vary significantly when the absolute values of the slopes of f(x) and g(x) in Eq. (A1) are similar. Suppose two lines to represent curves are chosen with slopes m_1 and m_2 such that

$$y = m_1x \tag{A2}$$

and

$$y = m_2x. \tag{A3}$$

Take m_2 less than m_1 and suppose both slopes have the same sign. Let z be the actual root of

$$m_1x = m_2x. \tag{A4}$$

Take x_0 to be initial approximation of z. If x is replaced by x_0 in the right-hand side of Eq. (A4), then

$$m_1x = m_2x_0, \tag{A5}$$

which can be solved by calculating x, in the

$$x_1 = m_2 x_0 / m_1 \tag{A6}$$

At this stage the tolerance can be written as $|x_1 - x_0|$ and the error size as $|x_1 - z|$. It is desirable to investigate the situation in which tolerance is less than a small number ε, and the error size is greater than ε. Symbolically, the following two relations are to be satisfied:

$$|x_1 - x_0| = |m_2 \; x_0/m_1 - x_0| < \varepsilon \tag{A7}$$

and

$$|x_1 - z \;| = |m_2 \; x_0/m_1 - 0| > \varepsilon \quad . \tag{A8}$$

Note that z is equal to zero in Eq. (A8), since the curves expressed in Eqs. (A2) and (A3) intersect at the origin [x = 0 satisfies Eq. (A4)].

Eq. (A7) may be re-written as

$$|x_1 - x_0| \;\; = |x_0| \; |\frac{m_2}{m_1} \; - 1| < \varepsilon$$

or

$$|x_1 - x_0 \;| = |x_0| \; (1 - \frac{m_2}{m_1}) < \varepsilon \quad .$$

Therefore

$$m_2/m_1 > 1 - \varepsilon/|x_0|. \tag{A9}$$

By substituting $1 - \varepsilon/|x_0|$ for m_2/m_1 in Eq. (A8), one obtains

$$|x_1 - z| > (1 - \frac{\varepsilon}{|x_0|} \;) \; |x_0| > \varepsilon \; ,$$

which may be simplified to

$$|x_0| - \varepsilon > \varepsilon \; .$$

Hence

$$|x_1 - z| > |x_0| > 2\varepsilon.$$

Therefore, if the seed x_0 is chosen such that $x_0 > 2\varepsilon$, then the calculated root x_1 will differ from x_0 by less than the tolerance ε without x being within a distance ε of the true root z. Thus, wherever the error size is greater than 2ε, the tolerance $|x_1 - x_0|$ is not a good measure of the error.

Take, for example, the lines

$$y = x \tag{A11}$$

and

$$y = \frac{999}{1000} x \quad . \tag{A12}$$

It is clear that the slopes of these two lines are very similar and that the point of intersection is (0,0). The actual root z of $x = \frac{999}{1000} x$ is 0.

Suppose a guess of 0.5 was made for the value of this root. Let this guess be called x_0. If x_0 is put in Eq. (A12), y takes the value 0.4995. If this value is put in the place of y in Eq. (A11), the first approximation x_1 is equal to 0.4995.

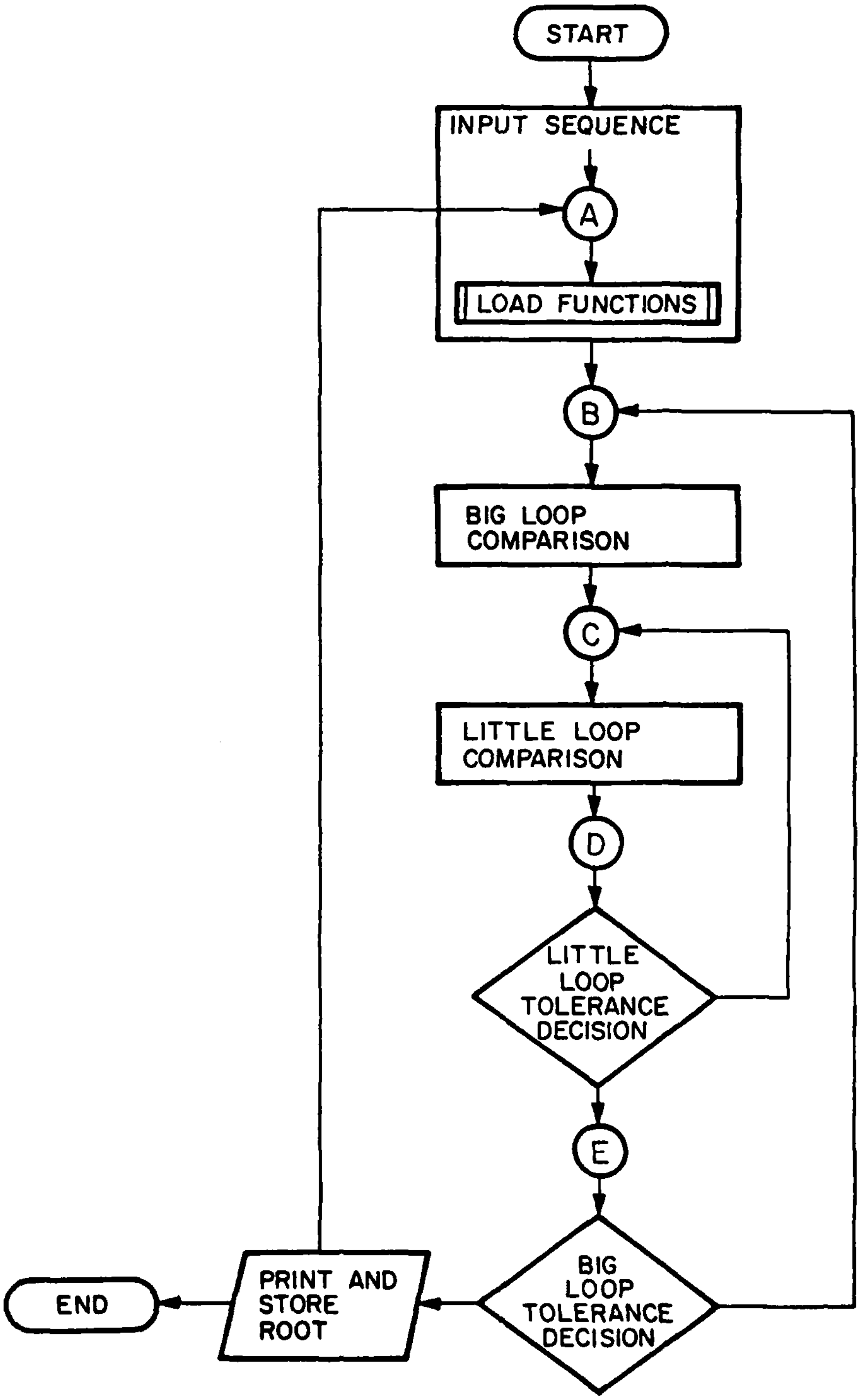
It can be seen that the difference $|x_0 - x_1| = 0.0005$. Certainly if ε were set at 1×10^{-3}, the difference $|x_0 - x_1| \leq \varepsilon$.

However, the difference $|z - x_0| = 0.4995$ which is not less than the predetermined tolerance. Hence, a rather simple example shows why tolerance is not always a good measure of error.

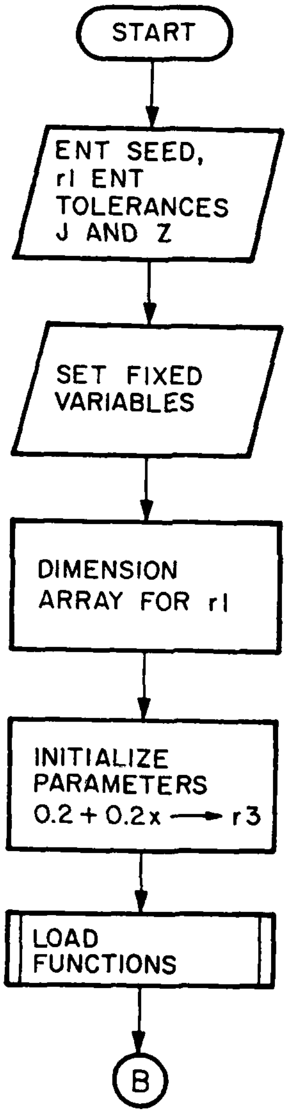
The author is aware that an argument has been presented only for linear equations at the origin. Linear equations were chosen because curves appear much like straight lines near the point of intersection. It is chosen that the equations have y-intercepts of zero in the examples for simplicity. A parallel argument holds for curves that intersect at a point other than the origin.

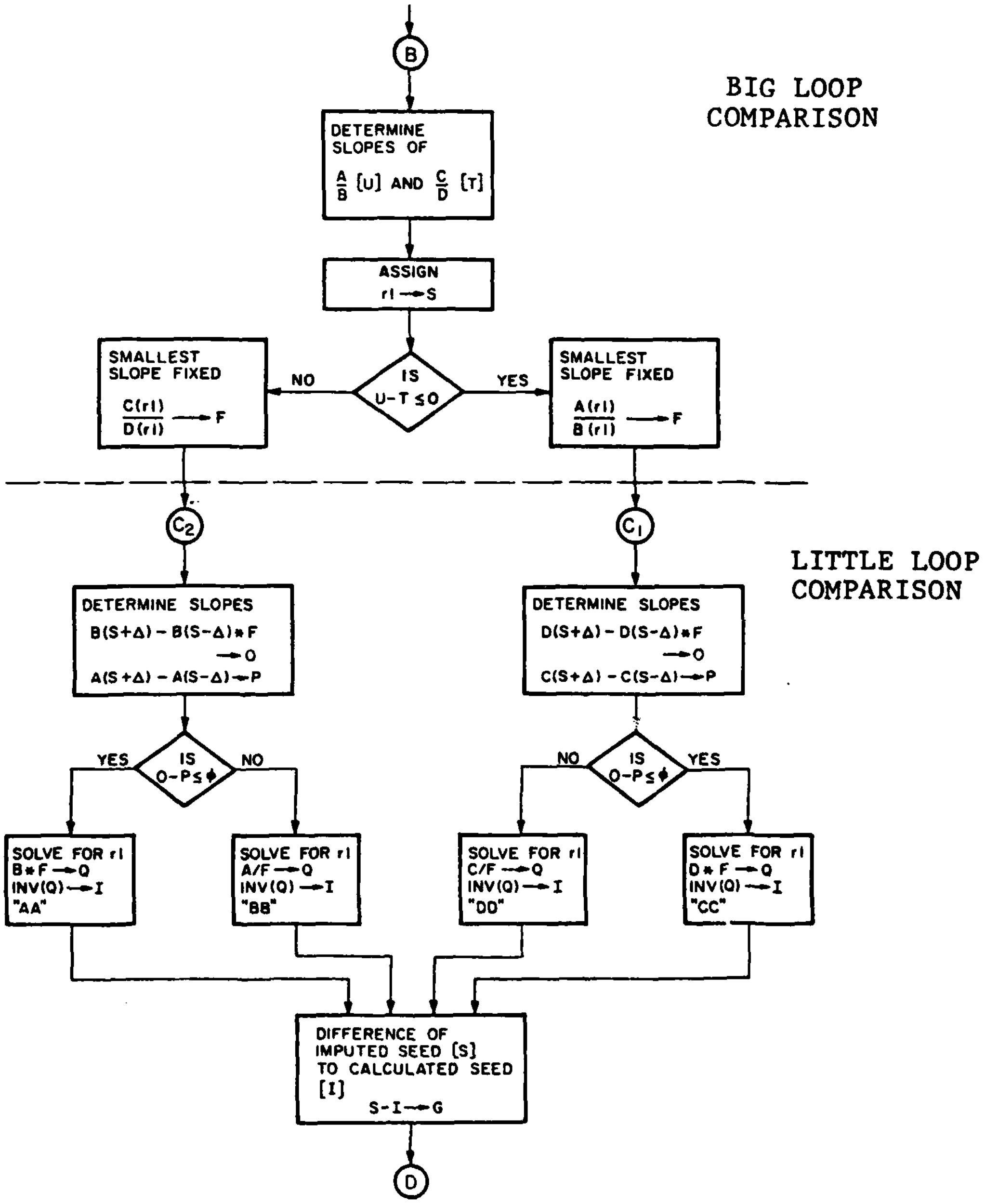
APPENDIX B

GENERAL FLOW DIAGRAM

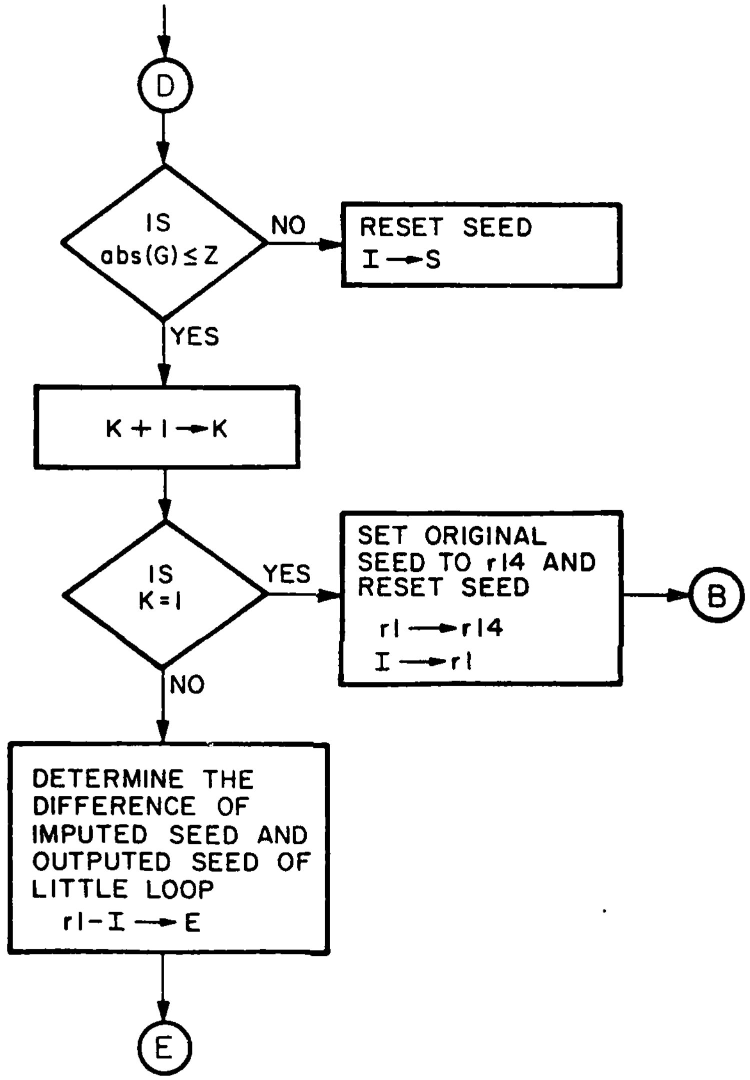


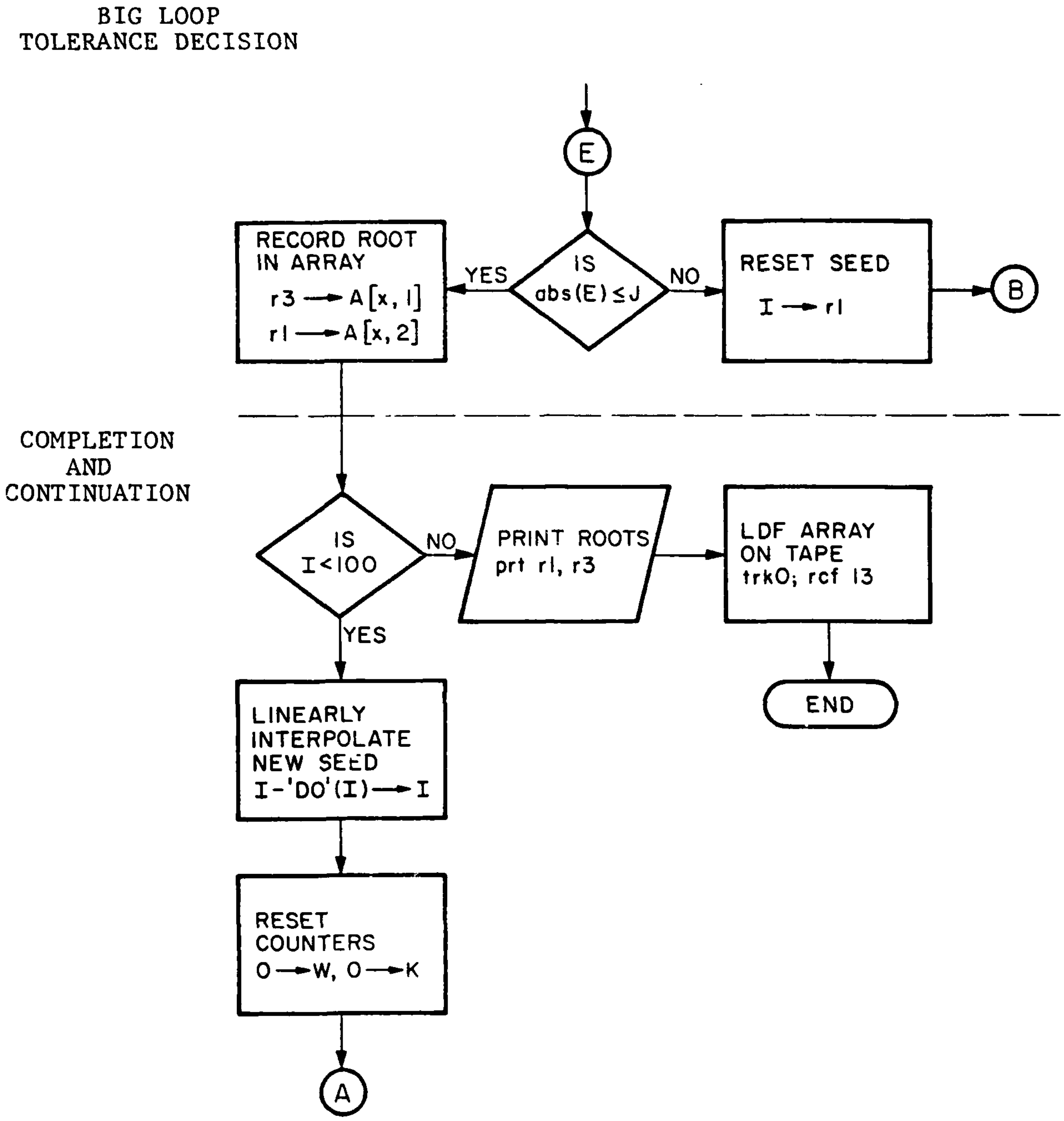
INPUT SEQUENCE


START
ENT SEED, rl ENT TOLERANCES J AND Z
SET FIXED VARIABLES
DIMENSION ARRAY FOR rl
INITIALIZE PARAMETERS 0.2 + 0.2x → r3
LOAD FUNCTIONS
B


B
BIG LOOP COMPARISON
DETERMINE SLOPES OF $\frac{A}{B}$ [U] AND $\frac{C}{D}$ [T]
ASSIGN rl → S
IS U−T ≤ 0
NO
YES
SMALLEST SLOPE FIXED $\frac{C(rl)}{D(rl)}$ → F
SMALLEST SLOPE FIXED $\frac{A(rl)}{B(rl)}$ → F
C2
C1
LITTLE LOOP COMPARISON
DETERMINE SLOPES B(S+Δ) − B(S−Δ)*F → O A(S+Δ) − A(S−Δ) → P
DETERMINE SLOPES D(S+Δ) − D(S−Δ)*F → O C(S+Δ) − C(S−Δ) → P
YES
IS O−P ≤ ϕ
NO
NO
IS O−P ≤ ϕ
YES
SOLVE FOR rl B*F → Q INV(Q) → I "AA"
SOLVE FOR rl A/F → Q INV(Q) → I "BB"
SOLVE FOR rl C/F → Q INV(Q) → I "DD"
SOLVE FOR rl D*F → Q INV(Q) → I "CC"
DIFFERENCE OF IMPUTED SEED [S] TO CALCULATED SEED [I] S−I → G
D

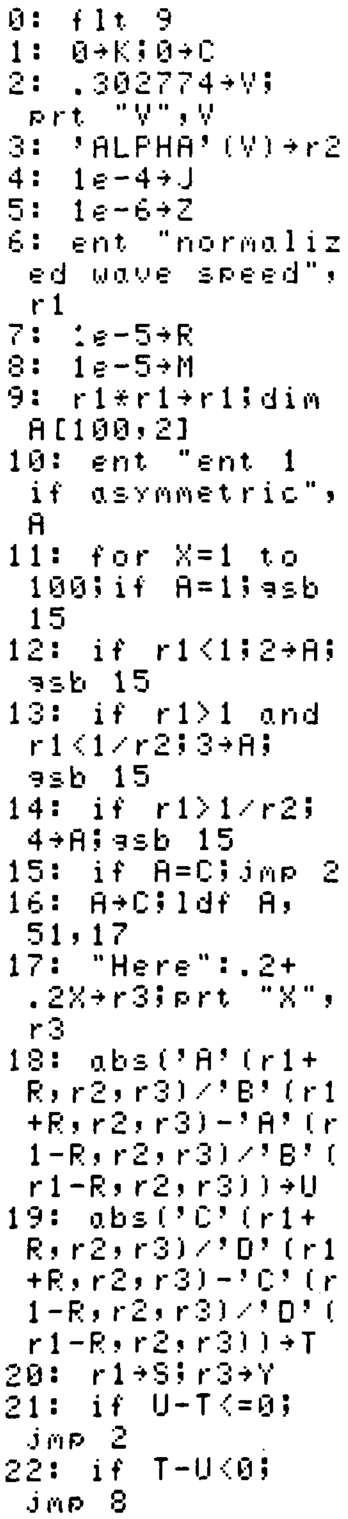
LITTLE LOOP TOLERANCE DECISION


D
IS abs(G) ≤ Z
NO
RESET SEED
I → S
YES
K + 1 → K
IS K = 1
YES
SET ORIGINAL SEED TO r14 AND RESET SEED
r1 → r14
I → r1
B
NO
DETERMINE THE DIFFERENCE OF IMPUTED SEED AND OUTPUTED SEED OF LITTLE LOOP
r1 - I → E
E


BIG LOOP
TOLERANCE DECISION
E
IS
abs(E) ≤ J
YES
NO
RECORD ROOT
IN ARRAY
r3 → A[x, 1]
r1 → A[x, 2]
RESET SEED
I → r1
B
COMPLETION
AND
CONTINUATION
IS
I<100
NO
YES
PRINT ROOTS
prt r1, r3
LDF ARRAY
ON TAPE
trkO; rcf 13
END
LINEARLY
INTERPOLATE
NEW SEED
I-'DO'(I) → I
RESET
COUNTERS
O → W, O → K
A

APPENDIX C

ITERATIVE HP PROGRAM



```
0: flt 9
1: 0→K;0→C
2: .302774→V;
 prt "V",V
3: 'ALPHA'(V)→r2
4: 1e-4→J
5: 1e-6→Z
6: ent "normaliz
 ed wave speed",
 r1
7: 1e-5→R
8: 1e-5→M
9: r1*r1→r1;dim
 A[100,2]
10: ent "ent 1
 if asymmetric",
 A
11: for X=1 to
 100;if A=1;gsb
 15
12: if r1<1;2→A;
 gsb 15
13: if r1>1 and
 r1<1/r2;3→A;
 gsb 15
14: if r1>1/r2;
 4→A;gsb 15
15: if A=C;jmp 2
16: A→C;ldf A,
 51,17
17: "Here":.2+
 .2X→r3;prt "X",
 r3
18: abs('A'(r1+
 R,r2,r3)/'B'(r1
 +R,r2,r3)-'A'(r
 1-R,r2,r3)/'B'(
 r1-R,r2,r3))→U
19: abs('C'(r1+
 R,r2,r3)/'D'(r1
 +R,r2,r3)-'C'(r
 1-R,r2,r3)/'D'(
 r1-R,r2,r3))→T
20: r1→S;r3→Y
21: if U-T<=0;
 jmp 2
22: if T-U<0;
 jmp 8
```

COMMENTS

Line 2:	Poisson's ratio (V)
Line 3:	Alpha in terms of Poisson's ratio. See line 47.
Line 4:	Big Loop Tolerance (J) Little Loop Tolerance (Z)
Line 6:	Original Seed (r1)
Lines 7 & 8:	Big Loop and Little Loop deltas
Lines 11-14:	Choosing the correct set of functions to be entered.
Line 17:	Initializing and incrementing parameter (r3)
Lines 18 & 19:	Determining the slopes of A/B and C/D

```
23: 'A'(S,r2,
 Y)→r8;'B'(S,r2,
 Y)→r9;r8/r9→F
24: abs(('D'(S+
 M,r2,Y)-'D'(S-
 M,r2,Y))*F)→O
25: abs('C'(S+M,
 r2,Y)-'C'(S-M,
 r2,Y))→P
26: if O-P<=0;
 jmp 2
27: jmp 2
28: -11→L;'D'(S,
 r2,Y)*F→Q;'CC'(
 S,r2,Y,Q)→I;
 gto "LL"
29: -10→L;'C'(S,
 r2,Y)/F→Q;'DD'(
 S,r2,Y,Q)→I;
 gto "LL"
30: 'C'(S,r2,Y)/
 'D'(S,r2,Y)→F
31: abs(('B'(S+
 M,r2,Y)-'B'(S-
 M,r2,Y))*F)→O
32: abs('A'(S+M,
 r2,Y)-'A'(S-M,
 r2,Y))→P
33: if O-P>=0;
 jmp 2
34: -5→L;'B'(S,
 r2,Y)*F→Q;'AA'(
 S,r2,Y,Q)→I;
 gto "LL"
35: -4→L;'A'(S,
 r2,Y)/F→Q;'BB'(
 S,r2,Y,Q)→I;
 gto "LL"
36: "LL":
37: S-I→G
38: if abs(G)<=Z
 ;K+1→K;gto "BL"
39: I→S;jmp L
40: "BL":if K=1;
 r1→r14;I→r1;
 gsb 18
41: r1-I→E
42: if abs(E)<=J
 ;gto "print"
43: I→r1;gsb 18
44: "print":r3→A
 [X,1];√I→A[X,
 2];prt "R",√I;
 I-'DO'(I)→r1;
 0→W;0→K;√I→r13;
 next X
45: ent "ldf
 #for Data",B
46: trk 0;rcf B,
 A[*];end
47: "ALPHA":ret
 (1-2V)/2(1-V)
48: "DO":I→r5
49: if X=1 or
 A=1;r5→r6;ret 0
50: r6-r5→r7;
 r5→r6;ret r7
```

COMMENTS

Lines
23 & 30: Fixing equation with smaller slope

Lines 24, 25,
31, & 32: Slope Decision in Little Loop

Lines
26 & 33: LL Slope comparison

Lines 28, 29,
34, & 35: Solving for variable with smaller slope

Line 38: Little Loop Tolerance Decision

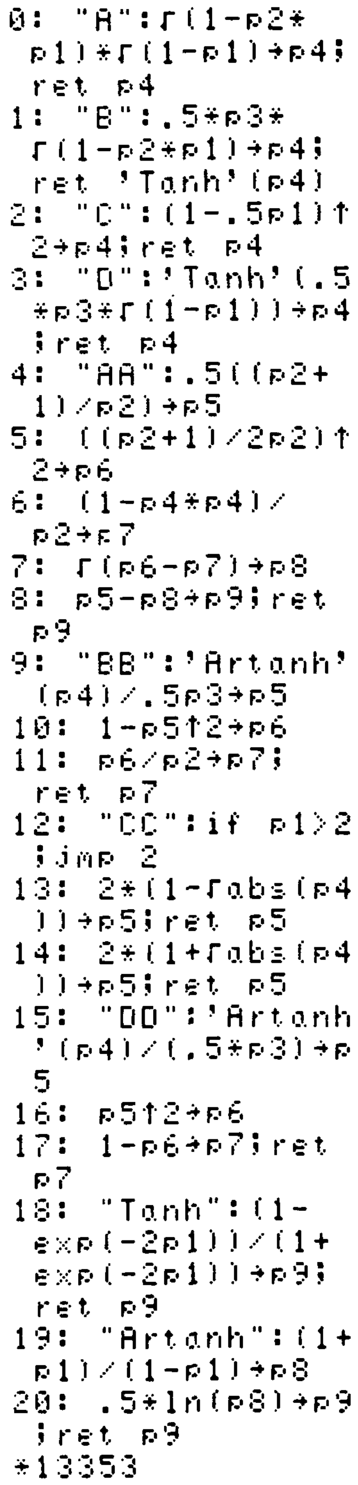
Line 42: Big Loop Tolerance Decision

Line 44: Prints roots (r1) and incremental parameter (r3). The subroutine 'DO' is a linear interpolation for new seed. See Lines 48-50.

Lines
45 & 46: Loads and stores array on tape

APPENDIX D

FUNCTIONS AND INVERSES OF ANTISYMMETRIC LAMB MODE

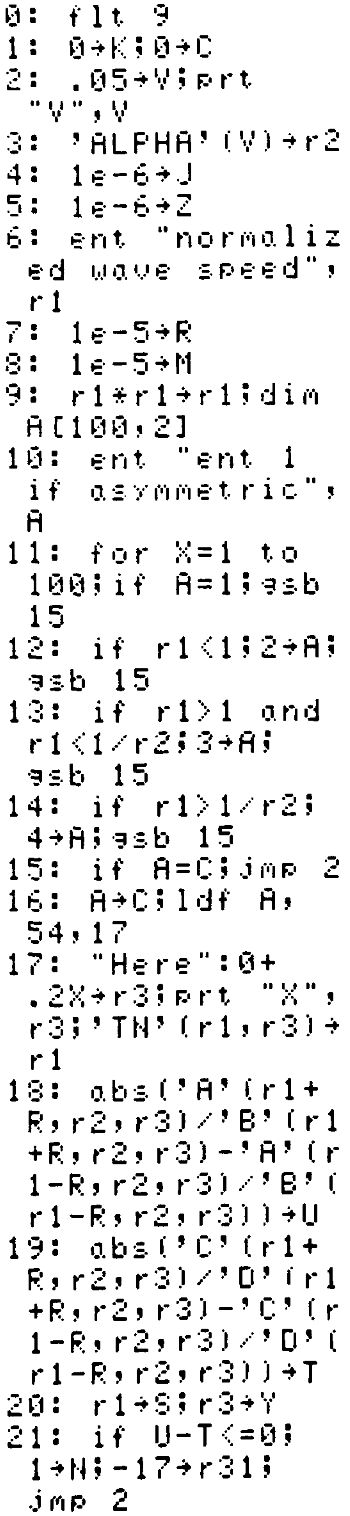


```
0: "A":√(1-p2*
 p1)*√(1-p1)→p4;
 ret p4
1: "B":.5*p3*
 √(1-p2*p1)→p4;
 ret 'Tanh'(p4)
2: "C":(1-.5p1)↑
 2→p4;ret p4
3: "D":'Tanh'(.5
 *p3*√(1-p1))→p4
 ;ret p4
4: "AA":.5((p2+
 1)/p2)→p5
5: ((p2+1)/2p2)↑
 2→p6
6: (1-p4*p4)/
 p2→p7
7: √(p6-p7)→p8
8: p5-p8→p9;ret
 p9
9: "BB":'Artanh'
 (p4)/.5p3→p5
10: 1-p5↑2→p6
11: p6/p2→p7;
 ret p7
12: "CC":if p1>2
 ;jmp 2
13: 2*(1-√abs(p4
 ))→p5;ret p5
14: 2*(1+√abs(p4
 ))→p5;ret p5
15: "DD":'Artanh
 '(p4)/(.5*p3)→p
 5
16: p5↑2→p6
17: 1-p6→p7;ret
 p7
18: "Tanh":(1-
 exp(-2p1))/(1+
 exp(-2p1))→p9;
 ret p9
19: "Artanh":(1+
 p1)/(1-p1)→p8
20: .5*ln(p8)→p9
 ;ret p9
*13353
```

Blank

APPENDIX E

ITERATIVE HP PROGRAM FOR $\nu \leq 10$



```
0: flt 9
1: 0→K;0→C
2: .05→V;prt
 "V",V
3: 'ALPHA'(V)→r2
4: 1e-6→J
5: 1e-6→Z
6: ent "normaliz
 ed wave speed",
 r1
7: 1e-5→R
8: 1e-5→M
9: r1*r1→r1;dim
 A[100,2]
10: ent "ent 1
 if asymmetric",
 A
11: for X=1 to
 100;if A=1;gsb
 15
12: if r1<1;2→A;
 gsb 15
13: if r1>1 and
 r1<1/r2;3→A;
 gsb 15
14: if r1>1/r2;
 4→A;gsb 15
15: if A=C;jmp 2
16: A→C;ldf A,
 54,17
17: "Here":0+
 .2X→r3;prt "X",
 r3;'TN'(r1,r3)→
 r1
18: abs('A'(r1+
 R,r2,r3)/'B'(r1
 +R,r2,r3)-'A'(r
 1-R,r2,r3)/'B'(
 r1-R,r2,r3))→U
19: abs('C'(r1+
 R,r2,r3)/'D'(r1
 +R,r2,r3)-'C'(r
 1-R,r2,r3)/'D'(
 r1-R,r2,r3))→T
20: r1→S;r3→Y
21: if U-T<=0;
 1→N;-17→r31;
 jmp 2
```

COMMENTS

Line 17: To avoid interfering with tan curve, 'TN' was designed to adjust resubmitted seed. See Lines 51-53.

```
22: if T-U<0;
 2→N;-10→r31;
 jmp 8
23: 'A'(S,r2,
 Y)→r8;'B'(S,r2,
 Y)→r9;r8/r9→F
24: abs(('D'(S+
 M,r2,Y)-'D'(S-
 M,r2,Y))*F)→O
25: abs('C'(S+M,
 r2,Y)-'C'(S-M,
 r2,Y))→P
26: if O-P<=0;
 jmp 2
27: jmp 2
28: -11→L;'D'(S,
 r2,Y)*F→Q;'CC'(
 S,r2,Y,Q)→I;
 gto "LL"
29: -10→L;'C'(S,
 r2,Y)/F→Q;'DD'(
 S,r2,Y,Q)→I;
 gto "LL"
30: 'C'(S,r2,Y)/
 'D'(S,r2,Y)→F
31: abs(('B'(S+
 M,r2,Y)-'B'(S-
 M,r2,Y))*F)→O
32: abs('A'(S+M,
 r2,Y)-'A'(S-M,
 r2,Y))→P
33: if O-P>=0;
 jmp 2
34: -5→L;'B'(S,
 r2,Y)*F→Q;'AA'(
 S,r2,Y,Q)→I;
 gto "LL"
35: -4→L;'A'(S,
 r2,Y)/F→Q;'BB'(
 S,r2,Y,Q)→I;
 gto "LL"
36: "LL":S-I→G;
 W+1→W;if W>500
 and 1=N;r14→I;-
 10→r31;0→W;gto
 43
```

COMMENTS

Line 36: Number of LL iterations (W)

```
37: if W>500
 and 2=N;r14→I;
 gto 43
38: if abs(G)<=Z
 ;0→W;K+1→K;gto
 "BL"
39: I→S;jmp L
40: "BL":if K=1;
 r1→r14;I→r1→S;
 jmp r31
41: r1-I→E
42: if abs(E)<=J
 ;gto "print"
43: I→r1→S;jmp
 r31-3
44: "print":r3→A
 [X,1];┌I→A[X,
 2];prt "R",┌I;
 I-'DO'(I)→r1;
 0→W;0→K;┌I→r13;
 next X
45: ent "ldf
 #for Data",B
46: trk 0;rcf B,
 A[*];end
47: "ALPHA":ret
 (1-2V)/2(1-V)
48: "DO":I→r5
49: if X=1 or
 A=1;r5→r6;ret 0
50: r6-r5→r7;
 r5→r6;ret r7
51: "TN":1+(π/
 r3)↑2→r60
52: if r60<r1;
 ret r60
53: ret r1
```